Journal of Computer and System Sciences 78 (2012) 1175–1184

Contents lists available at SciVerse ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Feature-based opinion mining and ranking

Magdalini Eirinaki *, Shamita Pisal [1], Japinder Singh [2]

Computer Engineering Department, San Jose State University, One Washington Square, San Jose, CA 95192, United States

ARTICLE INFO

Article history:
Received 25 March 2011
Received in revised form 22 July 2011
Accepted 17 October 2011
Available online 3 November 2011

Keywords:
Opinion mining
Feature ranking
Sentiment analysis
Semantic orientation
Search engine

ABSTRACT

The proliferation of blogs and social networks presents a new set of challenges and opportunities in the way information is searched and retrieved. Even though facts still play a very important role when information is sought on a topic, opinions have become increasingly important as well. Opinions expressed in blogs and social networks are playing an important role influencing everything from the products people buy to the presidential candidate they support. Thus, there is a need for a new type of search engine which will not only retrieve facts, but will also enable the retrieval of opinions. Such a search engine can be used in a number of diverse applications like product reviews to aggregating opinions on a political candidate or issue. Enterprises can also use such an engine to determine how users perceive their products and how they stand with respect to competition. This paper presents an algorithm which not only analyzes the overall sentiment of a document/review, but also identifies the semantic orientation of specific components of the review that lead to a particular sentiment. The algorithm is integrated in an opinion search engine which presents results to a query along with their overall tone and a summary of sentiments of the most important features.

1. Introduction

The proliferation of blogs and social networks presents a new set of challenges and opportunities in the way information is searched and retrieved. According to comScore, a marketing research company that provides marketing data and services to many of the Internet's largest businesses, out of the 1.1 billion people who actively use the Internet around the globe, 738 million are regular users of social networking sites – about 67% [1]. It further states that if the regular users of other social computing activities such as blogging are added the figure rises to 76%. It is thus clear that there exists vast amount of information in social networking sites such as blogs, review sites, social networking applications, etc.

This information can be leveraged for many purposes, including re-ranking and presenting the results of a search engine. A typical search engine works on the basis of keyword similarity: a user submits a keyword-based query and the search engine returns a list of items that are relevant to this query, as well as user ratings/reviews, if available. An important aspect of this type of search is related to the features of the product, which play a crucial role in the decision making process of the potential customer. It is these features that distinguish one product from other similar products from different brands. Most of the companies focus on a specific feature as their selling point. With the expanse of the e-commerce and the social networking sites, most of the people are using the Internet to check the reviews of products before buying them. They also want to keep themselves updated about any social or political issue in the neighborhood, in the state, in the country and

* Corresponding author.
E-mail address: magdalini.eirinaki@sjsu.edu (M. Eirinaki).
[1] The author is currently affiliated with Aruba Networks.
[2] The author is currently affiliated with Google Inc.

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2011.10.007

then across the globe. As the number of reviews has been increasing in a rapid pace, it becomes difficult for the end user to sort the helpful reviews from the ones that do not contain valuable information.

For example, the user may seek to buy a camera, or find a dentist at his/her area. Even though a search engine will return some results, the user needs to first filter out the ones relevant to his/her search, and then iterate through the numerous reviews/ratings for further details, including feedback on the items/services, useful features, etc. This online experience differs significantly from how a similar real-world search would take place. People have always depended on other people's opinion and experiences while buying products or selecting service providers. It is human nature to learn from others' experiences and an ideal search engine should reflect and satisfy this need.

In this work we leverage the information found online in various social networks and use it to create a friendlier search experience for the end user. We focus on analyzing opinions expressed by people in customer reviews, blogs, and social networking applications. We propose an opinion mining and ranking algorithm that first classifies a review as positive, negative or neutral but also identifies the product's more representative features and assigns overall "impression" weights to each one of them. In other words, it also classifies each feature as positive, negative, or neutral in various levels of importance and presents the most important ones to the end user. For example, if the user's search is for an "iPad", the opinions regarding the iPad are retrieved. In addition important features of iPad like "screen size", "applications", "touch interface", etc. are identified and extracted from such reviews. The algorithm summarizes and ranks the opinions about these product features by giving them scores.

The proposed algorithm is not limited to product search, but works for more general topic searches as well. For example for a political issue like "health care bill", it will retrieve positive, negative and neutral (factual) results about the bill. In addition it will also present a list of features of the health care bill like "public opinion", "coverage of seniors", "women health", etc. which people have expressed strong opinion about. We make the proposed algorithm the core of an opinion search engine, named AskUs. The proposed search engine can assist potential customers to make a wise decision based on the features being offered by each company. It can also help companies understand which feature of the product is well received by the audience and which is not.

The remainder of the paper is organized as follows. We provide an overview of the related work in Section 2. Section 3 presents an overview of the opinion search engine's system design. Section 4 presents the proposed opinion mining and ranking algorithms. The experimental results are discussed in Section 5, and we conclude in Section 6.

2. Related work

Many interesting works exist that focus on extracting the opinions from the customer reviews. Some works focus on performing opinion mining to identify the semantic orientation of a review overall, whereas others focus on identifying and extracting the opinion words that will determine the semantic orientation. This line of work further divides into those who focus on the opinion word identification and semantic orientation, and those who also employ features as an additional tool in representing the semantic orientation of a review. Our work is mostly related to the latter category, but we do provide an overview of related work in the other research areas as well.

In [2] the authors analyze and propose the semantic orientation of conjoined adjectives using a log-linear regression model to predict if the two adjectives joined by conjunction are of same or different orientation. In [3] the authors calculate the semantic orientation of words based on their semantic association with pre-determined positive and negative words. The work presented in [4] proposes an unsupervised method to extract syntactic structures that specify the orientation of clauses for domain oriented semantic analysis. Both [2] and [4] use conjunction rules to extract context-dependent opinion words from large corpora. In [5] the authors propose methods to determine the term subjectivity and term orientation using semi-supervised learning process while in [6] the orientation of the subjective terms is determined by utilizing the term definitions contained in the glossaries and dictionaries. Most of the aforementioned approaches, however, only lay stress on opinion words and do not consider the features. Moreover, they do not incorporate the proposed methodology in a broader opinion mining framework.

A few works exist that perform sentence-level sentiment analysis (i.e. sentiment analysis that is using words but is not extracting representative features) [7–9]. In [7] the opinion words are classified individually and then the polarity of the opinion sentence is calculated by combining the individual opinion word polarity while in [9] the sentiment of each sentence is analyzed by identifying the sentiment expressions and subject terms. Sometimes the opinions regarding the products may not be explicitly mentioned on the customer review sites but they exist in web blogs. Techniques to extract opinions contained in the blogs are proposed in [10]. Finally there exist a few product-ranking techniques based on opinion mining of product reviews for specific languages, such as Chinese [11]. This line of work performs sentence-level sentiment analysis, without focusing on the determination of representative features of the review.

Feature-based opinion summarization [12] allows the customer to drill down the chain of reviews pertaining to a specific feature. Various data mining techniques to summarize the opinions of the existing customers by predicting semantic orientation of the words are proposed in [13]. People often use different words or phrases to describe the same feature. Grouping these features is crucial for effective opinion summary. The words that describe a feature of the product are referred to as "feature expression". Grouping the feature expression used for a particular feature of a product is addressed in [14] by semi-supervised learning, namely Expectation Maximization (EM) and Naive Bayesian Expectation Maximization. The work of [15] presents multilevel latent semantic association for categorizing the product features. The Opinion Observer, proposed

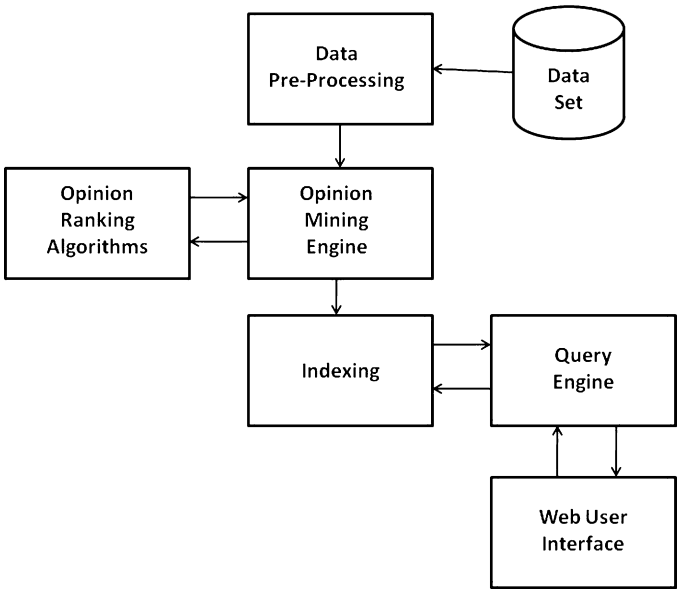

Fig. 1. System architecture.

in [16], provides visualizations that can help the potential customer to compare products by a mere glance at these visualizations. In [17] the authors propose a way to automatically mine the product features and the opinions by integrating the semi-structured and unstructured review sources. In this approach the mining results of the semi-structured reviews are treated as prior knowledge and used as a base to mine opinion and product features from the unstructured source using clustering based approach. It then finally integrates product features and opinions to form feature-opinion pairs using the Pointwise Mutual Information (PMI) statistics. Finally, the most similar approach to ours is that of [18], where the semantic orientation of a review sentence is determined by using the a summation as a function of opinion word, set of opinion words containing idioms, distance between feature and semantic orientation of each opinion word. Contrary to our work, however, the algorithm presented in [18] assigns the same weight to all the opinion words and do not follow the same approach as ours in identifying the features (they focus on idioms, whereas we focus on adjectives). Moreover, in our work, we have experimentally shown that assigning more weight to the opinion words expressed in the title of the review yields better results.

3. System architecture

Our motivation in this work is to develop an opinion search engine that will not only mine the opinions but will also extract useful information related to the item's features and use it to rate them as positive, neutral, or negative. This feature-based opinion mining will help the user focus on the features of the opinion/product he/she is interested in. The system will also give a brief summary of the search which will further aid the user. This will help the user spend less time going through reviews that do not add any value in decision-making. In this section we describe the system architecture of the opinion search engine AskUs, that incorporates all the processes from indexing and mining the data, to presenting them to the end user. In what follows we briefly outline the main components of our system, illustrated in Fig. 1.

Data preprocessing. The data from the data set is preprocessed so as to set the data in the format which is acceptable to the data processing algorithms. For example, the tag [t] is inserted at the beginning of the title to indicate that the sentence following [t] is the title of the review. Moreover, the reviews' file which corresponds to a particular product is split into text files containing one review each.

Opinion mining engine. The opinion mining engine comprises a POS (parts-of-speech) tagger module [19] and other utility modules used to process the text, such as, identifying the title of the review, and calculating the distance between a noun and its closest adjective.

Opinion ranking algorithm. The opinion ranking algorithm ranks the users' opinions based on the scores assigned to the derived features. These scores are used to decide the orientation of the opinion. The details of the ranking algorithm are included in Section 4.

Indexing. The extracted features and opinions are indexed using the indexer module to enable efficient retrieval and presentation from the user interface of the opinion search engine.

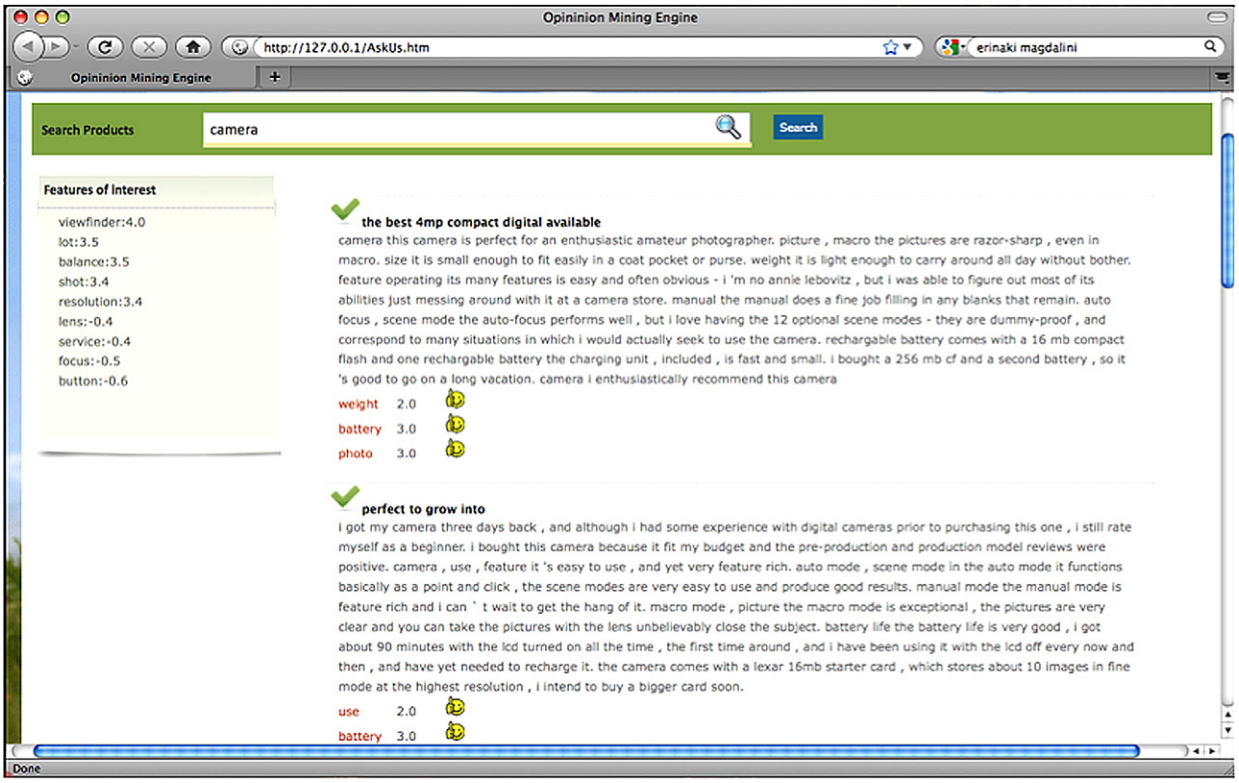

Fig. 2. AskUs interface.

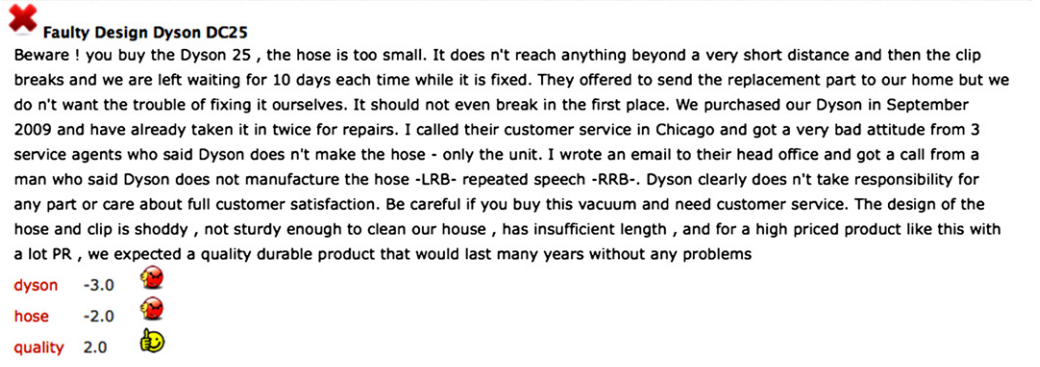

Fig. 3. A negative review.

Query engine. The query engine takes the query string as input, and preprocesses it. The preprocessing involves stopwords' removal and stemming. We have defined our own list of stopwords and used Porter's algorithm [20] for stemming.

User interface. The opinion search engine has a web user interface that enables the user to issue a query for the search engine. The user interface has a simple text box for the query input and a search button to submit the query to the backend engine. It displays the results in the form of positive, negative, and neutral opinions. It also displays the specifications or feature ratings of the queried input. Moreover, a search summary is displayed on the left side of the screen, that gives the use a quick overview of the search in terms of most important features of the product as mined from the reviews. An example of the interface is shown in Figs. 2 and 3. The added value of this interface is that it is very intuitive, making it easy for the end user to identify positive and negative reviews, as well as the distinguishing features and their respective ratings.

```
HighAdjectiveScores(reviews)
nouns_score_map<- { }
foreach review in reviews do
  Assign part of speech tags to the review
  Apply stemming
foreach line in the review do
foreach adjective in the line find the closest noun
noun_score_map[noun]++
potential_features_map<- { }
foreach noun in  noun_score_map
if noun_score_map[noun] > threshold
potential_features_map[noun] = threshold
return potential_features
```

Fig. 4. High Adjective Count algorithm.

4. Opinion mining and ranking algorithms

The core of the AskUs opinion search engine consists two algorithms, one for identifying and extracting the features that are deemed as the most important and characteristic of each review, and one that takes as input these features, assigns ranks to them and decides the final classification of the review as positive, neutral, or negative.

4.1. The High Adjective Count algorithm

The algorithm we propose to identify potential features is called the High Adjective Count (HAC) algorithm. In a nutshell, the main idea behind the algorithm is that the nouns for which reviewers express a lot of opinions are most likely to be the important and distinguishing features than those for which users don't express such opinions.

Instead of merely using the term frequency of the keywords, our algorithm starts by identifying the adjectives and nouns in the document collection.[3] The scores of the nouns are initialized to zero. Each adjective is associated with the noun to which it is the closest. This is the noun which the adjective is most likely to describe. For each such adjective, the score of the noun is increased by one. After processing all reviews in the document collection, the algorithm will have assigned scores for each of the nouns. In what follows, we'll refer to these as *opinion scores*. The opinion scores are used to rank the nouns such that, the higher ranked nouns will be the ones having more adjectives used to describe them. The scores are assigned such that there is one score per noun per item that is reviewed.

The ranking can then used to filter the nouns and identify potential features by selecting the nouns which have a score above a particular threshold. This threshold is a parameter of the algorithm and can be chosen based on experiments and human evaluation on different review data sets. The pseudocode of the aforementioned algorithm is shown in Fig. 4.

4.2. The Max Opinion Score algorithm

The second algorithm proposed is the one that ranks the extracted features using the opinion scores assigned in the previous step. This algorithm takes three inputs, each of which is described in the paragraphs below.

This first input is the list of adjectives which are used to express opinions. We refer to those as *opinion words*. In addition, for each of the adjectives in the list, we need to assign a score which indicates how positive or negative the opinion is. For example, "awesome" indicates a very strong positive opinion whereas "satisfactory" indicates an opinion which, while positive, is not as strong as that of "awesome". We have chosen to manually assign scores in the range $[-4, 4]$ to each of the opinion words. A negative score indicates a negative opinion, whereas a positive score indicates a positive opinion. A higher score indicates a stronger positive opinion than a lower score.

The second input to the algorithm is a list of inversion words. These are words like "not" which invert the sense of the opinion word. When these words occur in the left context of opinion words, they can invert the opinion sense. For example "not good" is a negative opinion. For this reason, when we are assigning scores to opinion words, we also maintain the left context, and if an inversion word appears in the context, we multiply the original score of the opinion word by -1.

The third input to this algorithm is the list of potential features. This can be identified using algorithms like the proposed HAC, or simpler, state-of-the-art ones, such as those that use word count (e.g. TF and TF-IDF).

We process each review sentence by sentence. For each sentence, we look at the opinion words and identify the feature closest to each one in the sentence. The score of the feature is the summation of the scores of the opinion words associated with that feature. The score of the identified features are further summed up to calculate the score of the review. For each feature we compute the average score per opinion word. This score is used to rank the features, based on the intuition that for positive reviews this will identify the features which reviewers like the most, and for negative reviews this will identify

[3] For this purpose we employ a POS (part of speech) tagger [19].

```
RankFeatures(adjective_scores, inversion_words, potential_features, reviews)
global_noun_scores = {}
global_noun_adjective_count = {}
foreach review in reviews
review_noun_scores = {}
review_noun_adjective_count = {}
  Apply pos tagging and stemming to the review
foreach line in review
    // Maintain a left context of two words
left_context = {}
line_score = 0
foreach word in line
if word in adjective_scores
score = adjective_score[word]
ifinversion_words in left_context
score = -1 * score
closest_noun = find_closest_noun(word)
review_noun_score[closest_noun] += score
    review_noun_adjective_count[closest_noun]++
global_noun_score[closest_noun] += score
    global_noun_adjective_count[closest_noun]++
line_score += score
        Update left_context
total_score = Sum of all scores in review_noun_score
total_adjectives = Sum of all scores in review_noun_adjective_count
avg_score = total_score / total_adjectives
ifavg_score> 0
    mark review as positive
else
    mark review as negative
avg_feature_score<- {}
Foreach noun in global_noun_score
avg_feature_score[noun] =
global_noun_score[noun] / global_noun_adjective_count[noun]
 Rank the features by avg_feature_score
```

Fig. 5. Maximum Opinion Score algorithm.

the ones which they are most unhappy with. The pseudocode illustrating the Max Opinion Score algorithm is shown in Fig. 5.

We should note that, for each review we score the features extracted from the title and body separately, as follows:

$$\text{Review Score} = \frac{\alpha \cdot \text{Title Score} + \text{Body Score}}{\alpha + 1} \tag{1}$$

where α is the *title weight coefficient*, Title Score $= \frac{\sum asc_t}{|a_t|}$, Body Score $= \frac{\sum asc_b}{|a_b|}$, asc_t and asc_b represent the adjective scores in the title and body respectively, and $|a_t|$ and $|a_b|$ represent the number of adjectives in the title and body respectively. The reasoning is that the title is most often a good summary that captures the overall mood of the reviewer and thus should be given a larger weight, as expressed by the title weight coefficient α. Nevertheless, α is a parameter of our algorithm and can be tuned appropriately depending on the data set on which the algorithm is applied.

4.3. Extensions for other social media

The proposed algorithm can be easily adapted to work with social media other than review sites, which provide us with a very specific structure of title/body of review. In the absence of the title signal the proposed algorithm will continue to function, at the expense of accuracy. However, this signal can be replaced by other indicative signals, depending on the specific application on which it is being applied. For instance, consider a social networking application like Facebook.[4] Even though the title is missing from the users' comments in such sites, we can exploit other signals that clearly indicate whether an opinion is positive or negative, such as the "Like" or the "Share" actions. Similarly a "retweet" is a signal showing a positive opinion on a specific tweet on the micro-blogging application Twitter.[5] Such signals can replace the title signal of Title Score in Eq. (1), whereas the weight coefficient α can be appropriately adjusted through experimentation.

[4] http://www.facebook.com.
[5] http://twitter.com.

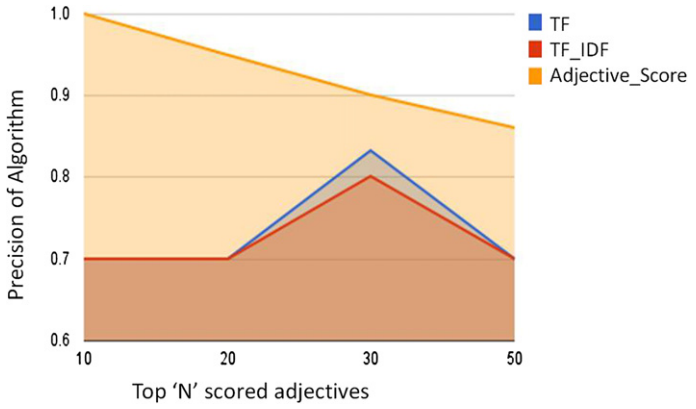

Fig. 6. Camera Data Set (80 reviews).

5. Experimental evaluation

For our experiments we used the product review datasets provided courtesy of Prof. Bing Liu.[6] We experimented with different data set sizes in order to measure several parameters of our system. More specifically, we performed three different sets of experiments. In the first line of experiments, we evaluated the performance of our opinion-based feature extraction algorithm (HAC), as compared to simple word count algorithms, such as TF and TF*IDF, all incorporated in our opinion mining algorithm. In the second line of experiments, we evaluated the significance of the title weight coefficient α. Finally, we measured the accuracy of our algorithms in correctly classifying reviews as positive or negative. We present the results of our experimental evaluation in the following subsections.

5.1. Evaluation of feature extraction algorithms

In this section we compare the performance of three algorithms used to extract representative features from a review, namely our proposed High Adjective Count (HAC), and the well-known TF and TF*IDF. We perform this by comparing the precision of the top-N results given by the three algorithms for various values of N. Precision in this context is defined as follows:

$$\text{Precision} = \frac{\#\text{ of relevant features}}{N} \tag{2}$$

An example of the comparison of the top-10 features for a camera review between the TF*IDF and the HAC algorithms is the following:

Common words – ['flash', 'battery', 'zoom', 'quality'].

6 Words in HAC only – ['features', 'pictures', 'cameras', 'camera', 'mode', 'images'].

6 Words in TF*IDF only – ['Picture', 'g3', 'canon', 'time', 'picture quality', 'software'].

It is clear that the words in HAC are better suited as potential features of a camera, whereas the TF*IDF algorithm has 'g3' and 'canon' which are brand terms, and 'time', which is a very general, irrelevant term. A human rater can thus assign a score of 7/10 to TF*IDF and 10/10 to HAC.

Figs. 6–9 present the precision for different number of features (N), for different review data sets. The comparison was performed by a human rater.

From the above experiments, it is clear that the HAC algorithm, which scores each noun found in the corpus based on the count of the number of adjectives in the corpus as the score, fares much better than the TF or TF*IDF algorithms. The only exception is the Vacuum Data Set, where TF*IDF gives better scores than the HAC algorithm for most of the sizes of N. We presume that this happened because this data set was the smallest one used in our experiments, and contained short reviews thus the inherent benefits of our algorithms could not be exploited. However, the algorithm's performance in the remaining datasets verifies our initial intuition that users comment on the features they care about using adjectives and the more often a noun is in the vicinity of an adjective, the more likely it is to be a representative feature of the item reviewed.

5.2. Thresholds for Computing Review Score

As part of our analysis, we are assigning a score to a review in the range $[-4, 4]$. Scores greater than zero are meant to indicate a positive review, and scores less than zero indicate a negative review. However, an important observation is that

[6] Retrieved from http://www.cs.uic.edu~liub/FBS/sentiment-analysis.html.

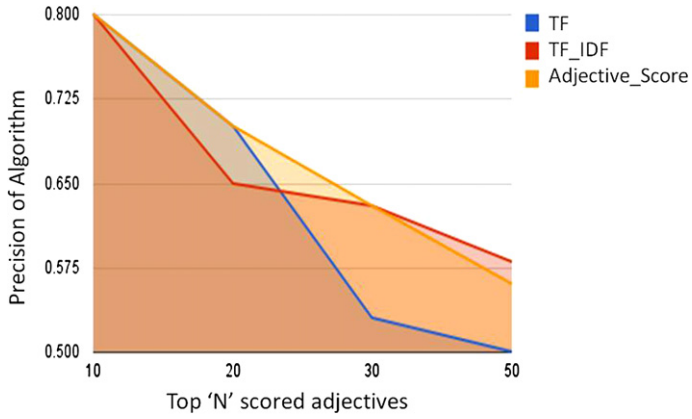

Fig. 7. DVD Player (small) Data Set (90 reviews).

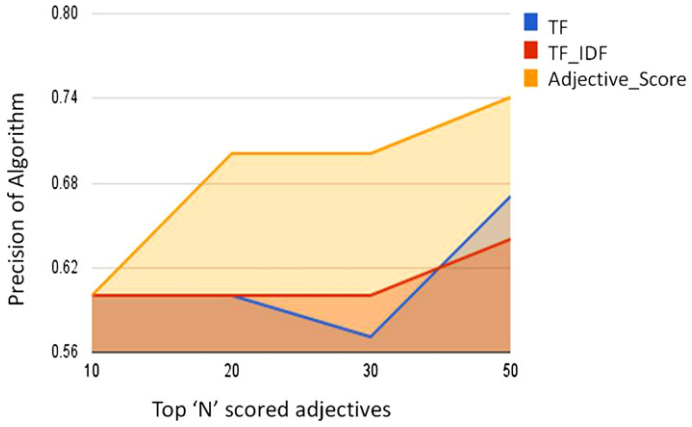

Fig. 8. DVD Player (large) Data Set (300 reviews).

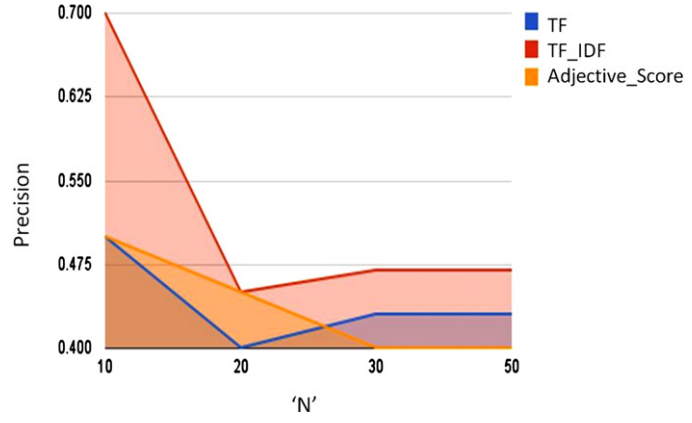

Fig. 9. Vacuum Data Set (40 reviews).

the title of a review is often a very accurate summary of the mood of the reviewer and should be given special consideration in assigning scores to the review. For this reason, we compute the score of a review as a linear combination of the title score and the body score, as discussed in Section 4.2.

An example is the following review. The first line (in bold) is the title. Without the title it is very hard to automatically derive whether the sentiment of the review is positive or negative as the reviewer uses hard to analyze expressions like 'you get what you pay for' to suggest his or her disappointment. By considering the sentiment of the title, however, we can figure out that the tone is negative: "***Do not buy this piece of junk***. *I purchased this unit* 3 *months back and I think the unit knew when my warranty expires. Picture, player it is more than* 90 *days and it does not show the picture no matter what I do. I can only hear*

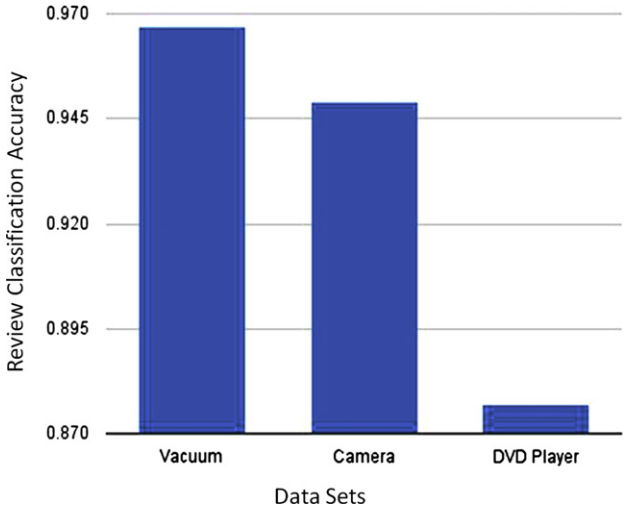

Fig. 10. Accuracy of the opinion mining algorithm.

the sound. I changed video cables, tried connecting to TV on input1 and input2 but no help. I think you get what you pay for stands true."

We experimented with several values of the title weight coefficient α to find out how the value of the scale factor can affect the accuracy of the classification of reviews as positive or negative. We started with a value of $\alpha = 1$ (i.e. equal weight to title and body of review), which turned an average accuracy (over all data sets) of 50%. Increasing the factor to $\alpha = 10$ yielded an average accuracy of 70%. However, further increasing the factor beyond 10 did not give us any improvement in the accuracy. Thus we concluded that a ratio of $10 : 1$ in the weight of the title vs. the body of a review improves the accuracy of the prediction.

5.3. Accuracy of review classification

For the various data sets, we have also computed the accuracy of our algorithms in classifying the reviews as positive or negative. In this context, accuracy is defined as follows:

$$\text{Accuracy} = \frac{\#\text{ of correct classifications}}{\text{total }\#\text{ of reviews}} \tag{3}$$

The correct classifications are determined using the information in a reference data set wherever available or using human evaluation where it was not. Fig. 10 shows the accuracy of the classified reviews for the various datasets we have experimented with. The results show that our algorithm succeeds to classify more than 87% of the reviews correctly in the worst case (DVD Player dataset), and has close to 97% accuracy in the best case. We should point out that the results of the combinations of our opinion mining algorithm with the TF and TF*IDF feature extraction algorithms are omitted since they yielded worse results than the HAC algorithm.

6. Conclusion

With the proliferation of social networking and e-commerce the information contained in the opinions/reviews expressed by the people has grown by leaps and bounds. In this work we present an opinion search engine system that incorporates two novel opinion mining algorithms. The opinions are based on features and the orientation of these opinions is also largely based on the features rather than a product as a whole. People seem to like/dislike a specific product because of some feature associated with the product. The proposed framework not only classifies a review as positive or negative, but also extracts the most representative features of each reviewed item, and assigns opinion scores on them. An initial experimental evaluation on several customer review data sets has shown that our algorithm achieves very high levels of accuracy.

Our plans for future work include experimenting with datasets from other social media, as discussed in Section 4.3. We also plan to further explore the idea of focusing on particular parts in a user's expressed opinion (e.g. the parts that are being most commented on) and extract features from there instead of the whole text.

References

[1] D. Winder, The importance of social mobility, Gemalto Review Magazine (2010) 9.

[2] V. Hatzivassiloglou, K.R. McKeown, Predicting the semantic orientation of adjectives, in: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, ACL'98, 1997, pp. 174–181.
[3] P.D. Turney, M.L. Littman, Measuring praise and criticism: Inference of semantic orientation from association, ACM Trans. Inf. Syst. 21 (2003) 315–346.
[4] H. Kanayama, T. Nasukawa, Fully automatic lexicon expansion for domain-oriented sentiment analysis, in: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP'06, 2006, pp. 355–363.
[5] B. Pang, L. Lee, Opinion mining and sentiment analysis, Found. Trends Inf. Retr. 2 (2008) 1–135.
[6] A. Esuli, F. Sebastiani, Determining the semantic orientation of terms through gloss classification, in: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, CIKM'05, 2005, pp. 617–624.
[7] S.-M. Kim, E. Hovy, Determining the sentiment of opinions, in: Proceedings of the 20th International Conference on Computational Linguistics, COLING'04, 2004.
[8] A. Meena, T.V. Prabhakar, Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis, in: Proceedings of the 29th European Conference on IR Research, ECIR'07, 2007.
[9] T. Nasukawa, J. Yi, Sentiment analysis: Capturing favorability using natural language processing, in: Proceedings of the 2nd International Conference on Knowledge Capture, K-CAP'03, 2003, pp. 70–77.
[10] W. Zhang, C. Yu, W. Meng, Opinion retrieval from blogs, in: Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management, CIKM'07, 2007, pp. 831–840.
[11] P. Tian, Y. Liu, M. Liu, S. Zhu, Research of product ranking technology based on opinion mining, Intelligent Computation Technology and Automation, vol. 4, International Conference, 2009, pp. 239–243..
[12] M. Hu, B. Liu, Mining opinion features in customer reviews, in: Proceedings of the 19th National Conference on Artificial Intelligence, AAAI'04, 2004, pp. 755–760.
[13] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'04, 2004, pp. 168–177.
[14] Z. Zhai, B. Liu, H. Xu, P. Jia, Grouping product features using semi-supervised learning with soft-constraints, in: Proceedings of the 23rd International Conference on Computational Linguistics, COLING'10, 2010, pp. 1272–1280.
[15] H. Guo, H. Zhu, Z. Guo, X. Zhang, Z. Su, Address standardization with latent semantic association, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'09, 2009, pp. 1155–1164.
[16] B. Liu, M. Hu, J. Cheng, Opinion observer: Analyzing and comparing opinions on the web, in: Proceedings of the 14th International Conference on World Wide Web, WWW'05, 2005, pp. 342–351.
[17] Q. Miao, Q. Li, R. Dai, An integration strategy for mining product features and opinions, in: Proceeding of the 17th ACM Conference on Information and Knowledge Management, CIKM'08, 2008, pp. 1369–1370.
[18] X. Ding, B. Liu, P.S. Yu, A holistic lexicon-based approach to opinion mining, in: Proceedings of the International Conference on Web Search and Web Data Mining, WSDM'08, 2008, pp. 231–240.
[19] Stanford log-linear part-of-speech tagger, 2010, http://nlp.stanford.edu/software/tagger.shtml.
[20] M. Porter, An algorithm for suffix stripping, Program 14 (3) (1980) 130–137.